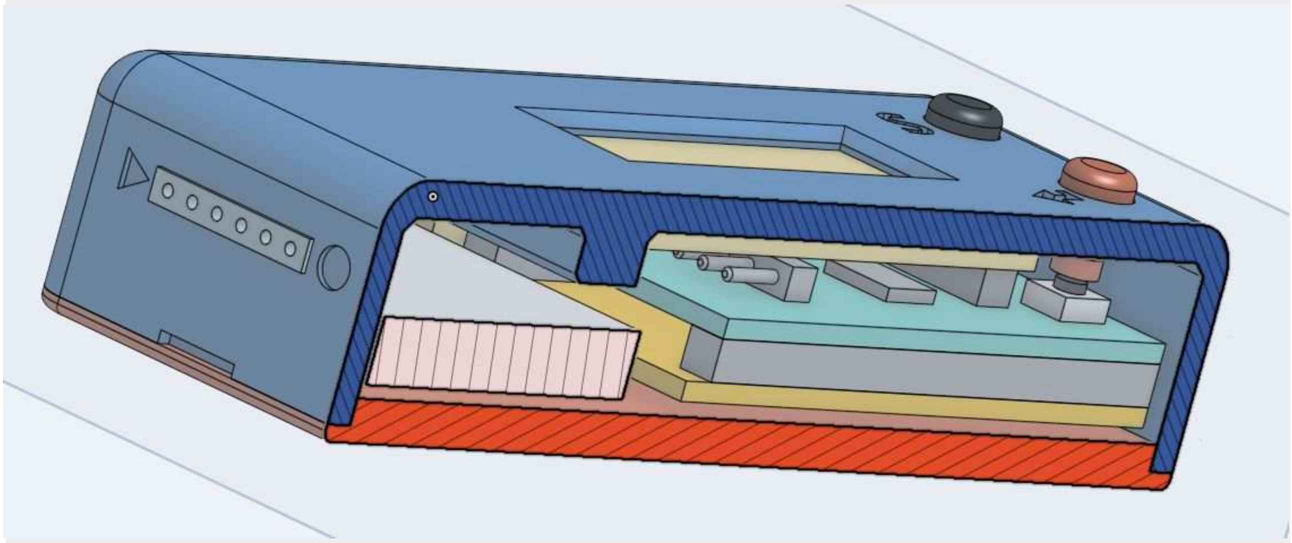
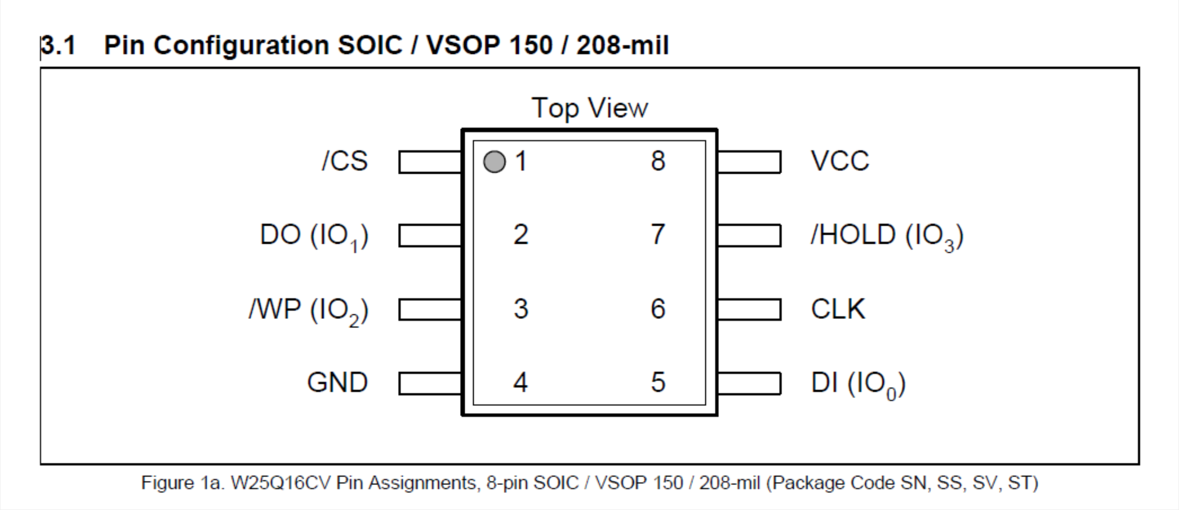
NOF a small file-OS for Flash

noForth button



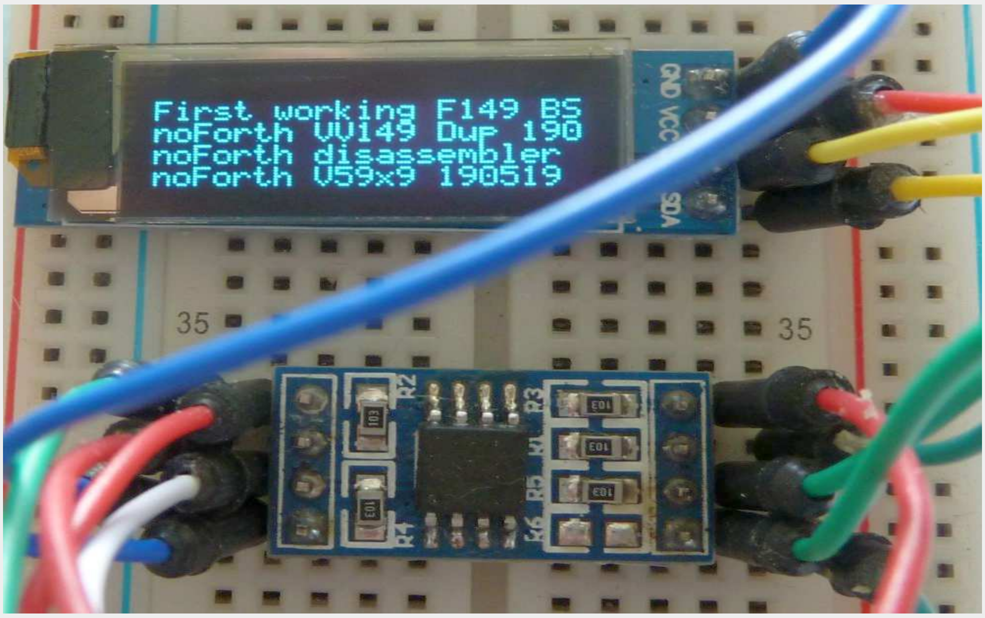
BSL programmer sketch



3.1 Pin Configuration SOIC / VSOP 150 / 208-mil

Figure 1a. W25Q16CV Pin Assignments, 8-pin SOIC / VSOP 150 / 208-mil (Package Code SN, SS, SV, ST)

W25Q16 to W25Q128



```
First working F149 BS
noForth VV149 Dup 190
noForth disassembler
noForth V59x9 190519
```

The memory structure of the Flash chip is leading:

- Reading is done in bytes
- Writing is done in 256 bytes sectors
- Erasure is done in 4096 bytes sectors

NOF structures

FET in first sector (**F**ile **E**ntry **T**able)	
0000 = FID	0C00 = DID

The format of FID (File ID's)

A FID-record is 4 cells (8-bytes)				
File	#Sector	Dhash	FHash	Flength
File-1	0010	73CB	834F	0030
File-2	0040	0000	753C	0020
Erased	0000	0000	0000	0000
Free	FFFF	FFFF	FFFF	FFFF

The format of DID (Directory ID's)

DID = 1-cell dir-hash & 14 bytes name		
Dir.	Dir-hash	Dir-naam
Dir-1	1DFD	NOFORTH
Dir-2	3195	EGEL
Dir-3	73CB	SOURCE
Free	FFFF	FF FF FF etc.

The format of a File

First 4096 Bytes sector	
FIT/AUX in two bytes	Filename = 30 bytes
File data	

What's the FIT and AUX

The FIT is the FIle Type, NOF now uses five of them:

```
0 = TI        - TI HEX file
1 = INTELHEX - Intel HEX file
2 = ASCII     - Plain text file
3 = FORTH     - Forth source file
4 = PICTURE   - Picture files
5 = BINARY    - General binary file
```

Space enough for a wealth of file types.

CRC16 CCITT

```
: CRC16     ( crc0 ch -- crc1 )
    8 lshift xor               \ Add 'ch' to crc0
    8 0 do
        dup 8000 and if        \ msb = '1'?
            2*                 \ Yes,
            1021 xor           \ Xor CRC CCITT polynomial
        else
            2*
        then
    loop ;
```

More on CRC, see **Project Forth Works** on Github.

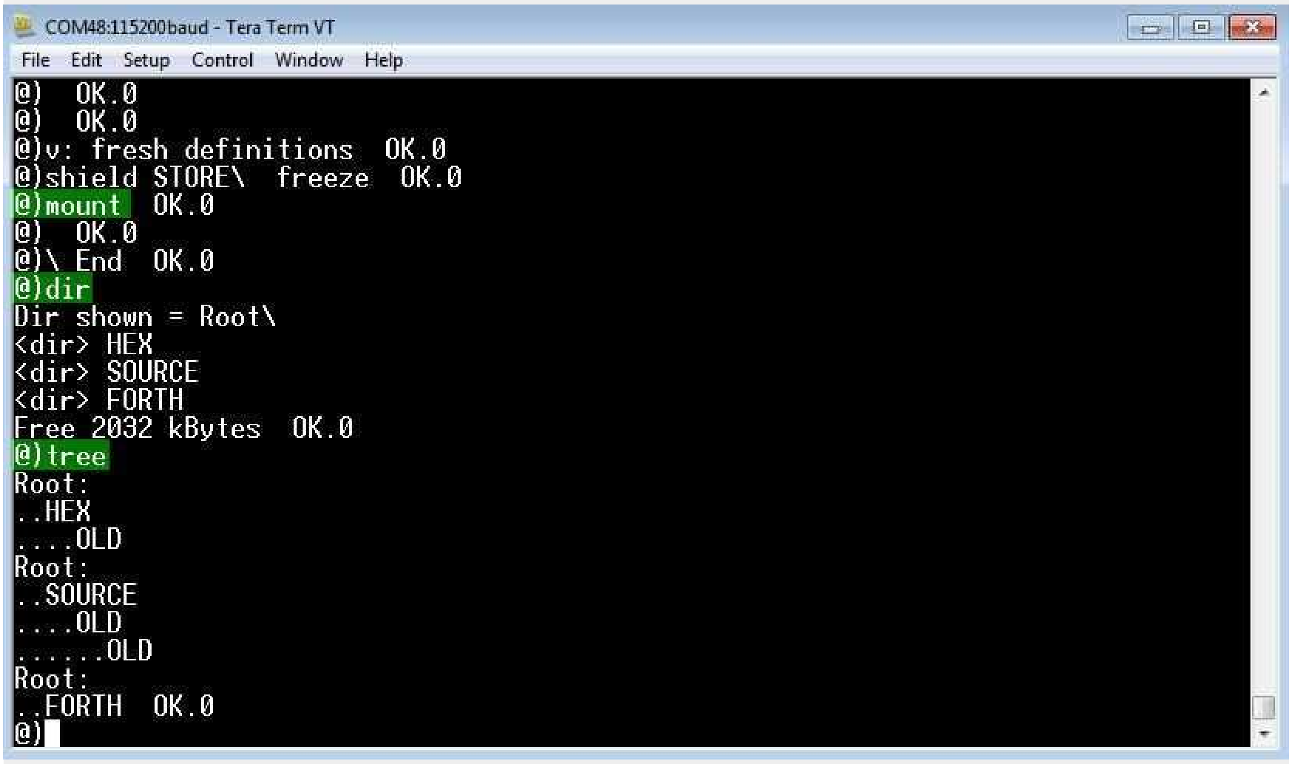
A few NOF functions

MOUNT	Connect an SPI Flash drive to NOF
MD	Make a new directory
CD	Choose and show the directory path
DIR	Show contents of the work directory
TREE	Show directory tree
VIEW	Display the contents of each file type
INCLUDE	Load a Forth source file name
DEL	Delete all files from the mounted disk
FORTH:	Add a named Forth source file
INTELHEX:	Add a named IntelHex file

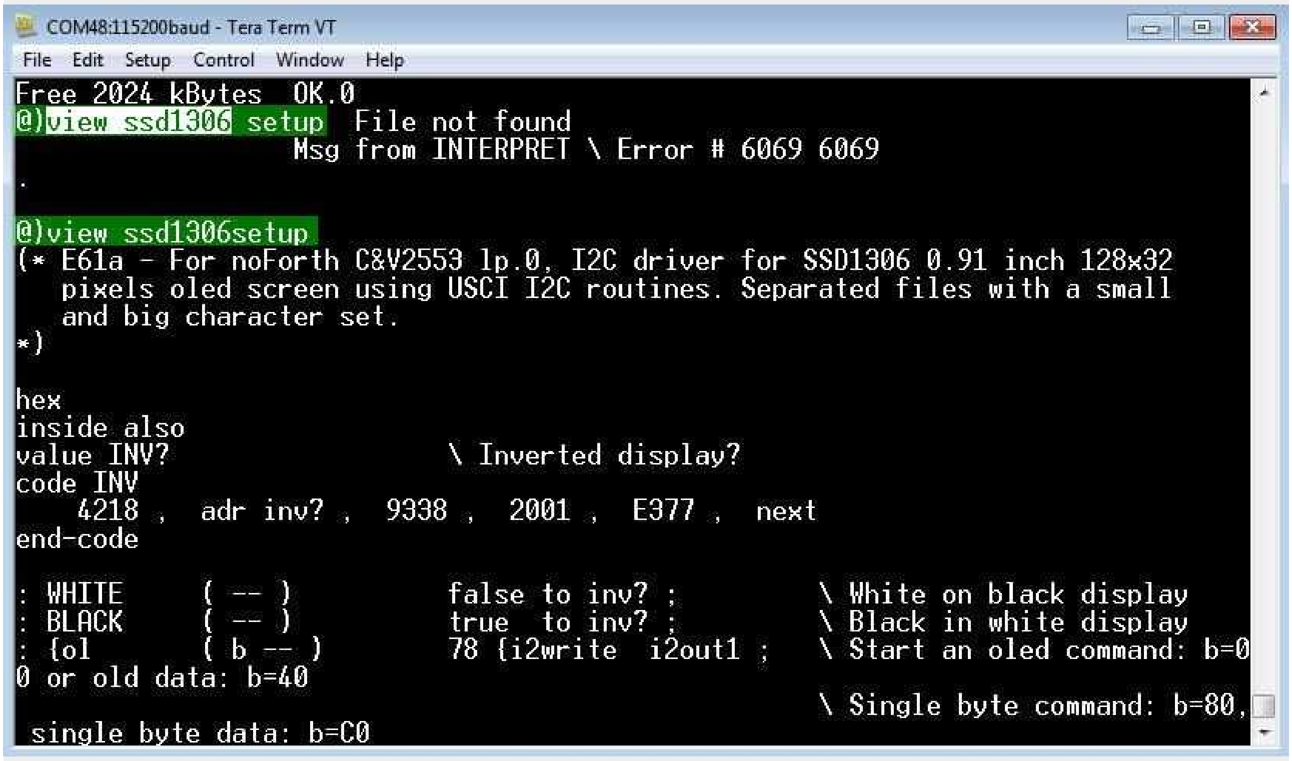
A few NOF code examples

```
: {FL          ( cmd -- )       8 322 *bic  FSPI-out ;
: FC@          ( da -- b )      03 {fread  fl} ;

\ Fill buffer +b from sa, the address of a 256 byte sector
: READ-SECTOR   ( sa +b -- )
    >r  03 {fl  addr-sector  r> 'buffer
    100 for  FSPI-in  over c!  1+  next  fl}  drop ;

: WRITE-ON    ( -- )    06 {fl fl} ; \ Enable write to Flash

\ Write buffer +b to sa address of a 256 byte sector
: WRITE-SECTOR  ( sa +b -- )
    >r  write-on  02 {fl  addr-sector  r> 'buffer
    100 for  count FSPI-out  next  drop  fl}  busy ;

\ Erase goes in 4 kByte sectors, so sa is a
\ 256 byte sector address within a 4 kByte sector!
: ERASE-SECTOR  ( sa -- )
    write-on  20 {fl addr-sector fl}  busy ;

0000 constant #FID  \ NOF block with File-ID's (384 files)
0C00 constant #DID  \ and Directory-ID's (64 directories)

\ Initialise FID and DID to next free location
\ Initialise first free sector location to #SECTOR
: MOUNT     ( -- )     \ Scan FET to intialise disk
    fspi-on  init-rwdata

    #FID  begin        \ Initialise FID to next free location
    dup 0 f@ -1 <> while
    dup to TMP 8 + repeat  #DID umin to 'FID

    #DID  begin        \ Initialise DID to next free location
    dup 0 f@ -1 <> while  10 +  repeat  1000 umin to 'DID

    #FID 'FID <> if    \ FID used?
        TMP 0 f@  TMP 6 + 0 f@ + \ Yes, calc. next free sector
    else  10  then     \ No, set to first free sector
    to #sector  ROOT ; \ Init. it, start in ROOT directory
```

NOF at work


COM48:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
@)  OK.0
@)  OK.0
@)v: fresh definitions  OK.0
@)shield STORE\  freeze  OK.0
@)mount  OK.0
@)  OK.0
@)\ End  OK.0
@)dir
Dir shown = Root\
<dir> HEX
<dir> SOURCE
<dir> FORTH
Free 2032 kBytes  OK.0
@)tree
Root:
..HEX
....OLD
Root:
..SOURCE
....OLD
......OLD
Root:
..FORTH  OK.0
@)
```


COM48:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
Free 2024 kBytes  OK.0
@)view ssd1306 setup  File not found
                  Msg from INTERPRET \ Error # 6069 6069
.

@)view ssd1306setup
(* E61a - For noForth C&V2553 lp.0, I2C driver for SSD1306 0.91 inch 128x32
   pixels oled screen using USCI I2C routines. Separated files with a small
   and big character set.
*)

hex
inside also
value INV?                  \ Inverted display?
code INV
    4218 ,  adr inv? ,  9338 ,  2001 ,  E377 ,  next
end-code

: WHITE     ( -- )          false to inv? ;        \ White on black display
: BLACK     ( -- )          true  to inv? ;        \ Black in white display
: {ol       ( b -- )        78 {i2write  i2out1 ;  \ Start an oled command: b=0
0 or old data: b=40
                                                   \ Single byte command: b=80,
 single byte data: b=C0
```

NOF include file

```
forth: Egel-demo
\ Include file for egel demo

include Asm
include FR5-usci0-i2c
include SD1306setup32
include Small-chars
include Thin-chars
include Fat-chars
include Graphic-chars
include Egel

\ End ;;;
```

NOF Intel-Hex file

```
#430FR intelhex: C2x55

:20180000C09D1A9E2E9E449A249EB69D5C9E549BB29428846E8448843E806C9E3E20E02E27
:08182000302F802F070010009B

:0E183200CF070006F4000608010204005155 1D
```

```
:20FF7E0000137EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFFFD
:20FF9E007EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF73
:20FFBE007EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF53
:20FFDE007EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF7EFF33
:02FFFE00529916

:00000001FF
```